CLAIMS:

What is claimed is:

1. A method in a data processing system for handling exception vectors by firmware, the method comprising:

identifying an exception vector to form an identified exception vector when control is passed from an operating system to firmware;

saving the identified exception vector to form a saved exception vector;

replacing the identified exception vector with a substitute vector; and

restoring the saved exception vector when control is returned to the operating system to form a restored exception vector, wherein the restored exception vector continues execution.

2. The method of claim 1 further comprising:

responsive to a processor, other than an associated processor associated with the save interrupt vector, generating an error for exception vector replaced by the substitute vector, placing the processor in slave loop until the saved exception vector is restored.

3. The method of claim 2, wherein the processor and the associated processor is identified using processor identification numbers.

4. The method of claim 2, wherein context for the processor in a slave loop is stored to form a stored context.

5. The method of claim 4, wherein the processor is restored by restoring the context restored by using the stored context and releasing the processor to the restored exception vector.

6. The method of claim 1, wherein the data processing system is a symmetric multi-processor data processing system.

7. A method in a data processing system for managing exception vectors, the method comprising:

receiving control from an operating system;

replacing an exception vector with substitute code; and

restoring the exception vector when control is returned to the operating system, wherein processors, other than a particular processor creating the exception vector, encountering the substitute code are suspended until control is returned to the operating system.

8. The method of claim 7, wherein the processors are suspended by placing the processors in a slave loop.

9. A data processing system for handling exception vectors by firmware, the data processing system comprising:

identifying means for identifying an exception vector to form an identified exception vector when control is passed from an operating system to firmware;

saving means for saving the identified exception vector to form a saved exception vector;

replacing means for replacing the identified exception vector with a substitute vector; and

restoring means for restoring the saved exception vector when control is returned to the operating system to form a restored exception vector, wherein the restored exception vector continues execution.

10. The data processing system of claim 9 further comprising:

generating means for generating an error for the exception vector replaced by the substitute vector in response to a processor, other than an associated processor associated with the save interrupt vector, and placing the processor in slave loop until the saved exception vector is restored.

11. The data processing system of claim 10, wherein the processor and the associated processor is identified using processor identification numbers.

12. The data processing system of claim 10, wherein context for the processor in a slave loop is stored to form a stored context.

13. The data processing system of claim 12, wherein the processor is restored by restoring the context restored by using the stored context and releasing the processor to the restored exception vector.

14. The data processing system of claim 9, wherein the data processing system is a symmetric multi-processor data processing system.

15. A data processing system for managing exception vectors, the data processing system comprising:

receiving means for receiving control from an operating system;

replacing means for replacing an exception vector with substitute code; and

restoring means for restoring the exception vector when control is returned to the operating system, wherein processors, other than a particular processor creating the exception vector, encountering the substitute code are suspended until control is returned to the operating system.

16. The data processing system of claim 15, wherein the processors are suspended by placing the processors in a slave loop.

17. A computer program product in a computer readable medium for handling exception vectors by firmware, the computer program product comprising:

first instructions for identifying an exception vector to form an identified exception vector when control is passed from an operating system to firmware;

second instructions for saving the identified exception vector to form a saved exception vector;

third instructions for replacing the identified exception vector with a substitute vector; and

fourth instructions for restoring the saved exception vector when control is returned to the operating system to form a restored exception vector, wherein the restored exception vector continues execution.

18. The computer program product of claim 17 further comprising:

fifth instructions for generating an error for exception vector replaced by the substitute vector in response to a processor, other than an associated processor associated with the save interrupt vector, and placing the processor in slave loop until the saved exception vector is restored.

19. The computer program product of claim 18, wherein the processor and the associated processor is identified using processor identification numbers.

20. The computer program product of claim 18, wherein context for the processor in a slave loop is stored to form a stored context.

21. The computer program product of claim 20, wherein the processor is restored by restoring the context restored by using the stored context and releasing the processor to the restored exception vector.

22. The computer program product of claim 17, wherein the data processing system is a symmetric multi-processor data processing system.

23. A computer program product in a computer readable medium for managing exception vectors, the computer program product comprising:

first instructions for receiving control from an operating system;

second instructions for replacing an exception vector with substitute code; and

third instructions for restoring the exception vector when control is returned to the operating system, wherein processors, other than a particular processor creating the exception vector, encountering the substitute code are suspended until control is returned to the operating system.

24. The computer program product of claim 23, wherein the processors are suspended by placing the processors in a slave loop.